
TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: 2612R011 – Elektronické informační a řídicí systémy

Kalibrační pracoviště pro měření průtoku

Calibration stand for flow measurement

Bakalářská práce

Autor: **Vít Bílek**

Vedoucí práce: Ing. Lubomír Slavík, Ph.D.

Konzultant: Ing. Josef Grosman

V Liberci 15. 5. 2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Vít Bílek**
Osobní číslo: **M11000145**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektronické informační a řídicí systémy**
Název tématu: **Kalibrační pracoviště pro měření průtoku**
Zadávající katedra: **Ústav mechatroniky a technické informatiky**


Z á s a d y p r o v y p r a c o v á n í :

1. Prostudujte problematiku jednočipových mikroprocesorů Silabs.
2. Navrhněte software přípravku pro sběr frekvenčních výstupů z minimálně 8 průtokoměrů.
3. Navrhněte software do PC pro porovnání měřených průtokoměrů s etalonem.
4. Oživte systém ověření a kalibrace průtokoměrů včetně nastavení průtoku.


Rozsah grafických prací: dle potřeby dokumentace
Rozsah pracovní zprávy: 30–40 stran
Forma zpracování bakalářské práce: tištěná/elektronická
Seznam odborné literatury:

- [1] Ďaďo, S., Kreidl, M.: Senzory a měřicí obvody. Vydavatelství ČVUT Praha 1999
- [2] Ďaďo S., Bejček L., Platil A.: Měření průtoku a výšky hladiny, BEN - technická literatura, Praha 2005, 1.vydání, ISBN80-7300-156-X
- [3] Pinker, J.: Mikroprocesory a mikropočítače. 1. vyd. Praha: BEN - technická literatura, 2004, 159 s. ISBN 80-730-0110-1.
- [4] Katalogové listy mikroprocesorů Silabs.

Vedoucí bakalářské práce: Ing. Lubomír Slavík, Ph.D.
Ústav mechatroniky a technické informatiky
Konzultant bakalářské práce: Ing. Josef Grosman
Ústav mechatroniky a technické informatiky
Datum zadání bakalářské práce: 10. října 2013
Termín odevzdání bakalářské práce: 16. května 2014


prof. Ing. Václav Kopecký, CSc.
děkan

L.S.


doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2013

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum:

Podpis:

Poděkování

Touto cestou bych rád poděkoval vedoucímu bakalářské práce Ing. Lubomíru Slavíkovi, Ph.D za ochotu a odborné rady při konzultacích.

Dále bych chtěl poděkovat konzultantovy Ing. Josefu Grosmanovi, za jeho odborné rady v oblasti mikroprocesorů.

Rád bych také poděkoval svým rodičům, za to, že mi umožnili studovat a za zázemí, které mi pro studium vytvořili.

Abstrakt

Tato bakalářská práce se zabývá ověřováním přesnosti a kalibrací měřičů průtoku kapalin. Cílem této bakalářské práce je software a úprava hardware elektroniky měřicí ústředny pro kalibrační linku a dále pak software do řídicího počítače, umožňující ověřování a kalibraci průtokoměrů.

Kalibrační linka je vybavena dvěma etalonovými průtokoměry a umožní připojení až 6 měřených průtokoměrů. Všechny měřené průtokoměry jsou vybaveny pulzními výstupy, přičemž kmitočet těchto pulzů je přímo úměrný průtoku. Tyto výstupy jsou přivedeny k měřicí ústředně, dále zpracovávány a pomocí sériového rozhraní RS232 jsou naměřené hodnoty posílány do počítače, kde se následně vyhodnocují. Nastavení průtoku je provedeno řízením frekvenčního měniče čerpadla, opět pomocí linky RS232.

V počítači se porovnají naměřené hodnoty nekalibrovaných průtokoměrů s hodnotami etalonů a vyhodnotí se přesnost měřených průtokoměrů.

V případě požadavku je možno provést kalibraci průtokoměrů.

Klíčová slova: průtokoměrná linka, kalibrace, etalon, C, C#, impulsní číslo, RS232

Abstract

This bachelor thesis is concerned with verifying of the accuracy and calibration of liquid flow measuring. Software and hardware modification of electronics logger for calibration line are the main task of this thesis. Software of control computer which provides the verification and calibration of flow measuring units is next task. The calibration line is equipped by two etalons and allows connection of up to 6 flow measuring units. All measured flow meters are equipped with pulse outputs. The frequency of these pulses is directly proportional to flow.

These outputs are connected to a data logger and further processed using a serial interface RS232. Measured values are sent to the computer and then evaluated.

Setting of the flow is allowed by controlling of the frequency changer of the water pump via the RS232. Measured values of uncalibrated flow meters are compared with the values of the etalon and the accuracy of the measured flow meters is evaluated. The calibration of flow meters can be done on the request.

Keywords: flow bench, calibration, standard, C, C#, pulse number, RS 232

Obsah

Abstrakt.....	5
Abstract.....	6
Obsah.....	7
Seznam ilustrací.....	8
Seznam vzorců.....	8
Úvod.....	9
1. Měření průtoku.....	11
1.1 Objemový průtok.....	11
1.2 Hmotnostní průtok.....	11
1.3 Měření v otevřeném kanálu.....	11
1.4 Měření v uzavřeném kanálu.....	12
2. Kalibrace průtoků.....	13
2.1 Kalibrace.....	13
2.2 Kalibrace přes váhu.....	13
2.2.1 Statické vážení.....	14
2.2.2 Dynamické vážení.....	15
2.3 Oběhová kalibrace.....	16
3. Praktická část.....	17
3.1 Kalibrační linka.....	17
3.2 Měřicí deska.....	19
3.2.1 Čítače a časovače.....	21
3.2.2 Sériový port.....	23
3.2.3 Měřicí program.....	26
3.3 Obslužný program.....	31
3.3.1 Grafické rozhraní.....	32
3.3.2 Informace o měřidlech.....	35
3.3.3 Měření.....	35
3.3.4 Komunikace s měřicí deskou.....	37
3.3.5 Zpracování dat.....	38
3.3.6 Ověřování.....	40
3.3.7 Kalibrace.....	42
3.4 Komunikace s průtokoměrem.....	44
3.5 Řízení oběhového čerpadla.....	45
Závěr.....	47
Seznam použité literatury.....	48
Seznam příloh.....	49

Seznam ilustrací

Metoda kalibrace statickým vážením.....	14
Metoda kalibrace dynamickým vážením.....	15
Metoda oběhové kalibrace.....	16
Kalibrační linka - blokové schéma.....	17
Program měřicí desky – blokové schémay.....	20
Registr TMOD.....	22
Odesílací protokol.....	25
Funkce přerušení UART – blokové schéma.....	26
Měřicí program – blokové schéma.....	27
Obslužný program – blokové schéma.....	31
Form, grafické rozhraní.....	33
Informace o měřidlech.....	35
Ověřování průtokoměrů.....	40
Kalibrace průtokoměrů.....	42
Protokol komunikace s průtokoměrem.....	44
Schéma zapojení a návrh DPS zesilovače pro řízení čerpadla.....	45

Seznam vzorců

Výpočet objemového průtoku.....	11
Výpočet hmotnostního průtoku.....	11
Výpočet modulační rychlosti RS232.....	24
Výpočet kalibrační konstanty new K.....	43
Výpočet kalibrační konstanty new P.....	43
Výpočet kontrolního součtu přenosového protokolu SILMPLE.....	44
Definice zesílení.....	45
Návrh rezistorů pro neinvertující zesilovač s OZ.....	45

Úvod

V dnešní době je měření průtoků kapalných a plyných látek velice důležité. Využívá se v mnoha aplikacích, zejména v mnoha případech spjatých s automatizační technikou a průmyslem.

Průtok se měří například v energetickém průmyslu v přehradách a vodních dílech, dále ve zdravotnictví, v potravinářském průmyslu, chemickém průmyslu. Stejně tak se měření průtoků využívá k měření odběru pitné vody a zemního plynu v domácnostech, k měření odebraného množství pohonných hmot u čerpacích stanic, atd.

Je zřejmé, že měření průtoků musí být velice přesné. Rozdíl mezi správně a chybně určeným průtokem je veliký problém, protože zde může vzniknout finanční zisk, nebo finanční ztráta. Na špatném určení průtoků mohou záviset i lidské životy. Například při špatném dávkování léků v nemocnicích hrozí poškození lidského zdraví, v nejhorším případě smrt pacienta. Při chybném měření přítoku vody do přehrad hrozí obecné ohrožení způsobené například špatným určením povodňového stupně při povodních. Při takovémto pochybení může dojít k přetečení, nebo protržení přehrady.

Cílem této bakalářské práce je uvést do provozu linku určenou ke kalibraci průtokoměrů, která by měla sloužit jako výukový prostředek. Hlavním úkolem linky je sbírat a zpracovávat data z bytových a průmyslových průtokoměrných měřidel umístěných na průtokoměrné lince. Na základě obdržených dat linka určí impulsní konstanty a vypočítá kalibrační koeficienty pro následné zkalibrování průtokoměrných měřidel. Kalibrace by měla zajistit, aby měřidla měřla v toleranci uvedené jejich výrobcem.

Průtokoměrná linka byla vyrobena pro školní laboratoře Technické univerzity v Liberci v roce 2011. Na lince může být najednou usmístěno až šestnáct průtokoměrů, z nichž bude alespoň jeden etalon, podle kterého se budou kalibrovat ostatní měřicí přístroje. Každý z nich generuje signál o určité frekvenci. Ten odpovídá okamžitému průtoků. Průtok se dá z pravidla určit z frekvence tak, že se frekvence vynásobí impulsní konstantou. Ta bývá uvedena buď v impulsech na litr, impulsech na metr

krychlový nebo v litrech na impuls.

Signály generované průtokoměry jsou přivedeny na měřicí desku. Jako první se provede zpracování signálů, kdy se sinusové signály převedou na obdelníkový signál 0 až 3,3 V. Vstupní obvody převedou harmonický signál diskretní log „0“ a „1“. Tyto signály jsou přivedeny na Porty mikroprocesoru. Mikroprocesor sleduje změny signálu a v příslušný okamžik zaznamená příchozí puls. Podle toho, jak rychle přicházejí pulsy za sebou, určuje frekvence příchozích signálů. Na vyžádání poskytuje naměřená data obslužnému programu v PC.

Veškeré zpracování dat se provádí v počítačovém programu. V tomto programu je nutné nastavit dobu trvání měřicího procesu a určit pozici etalonového průtokoměru. Po nastavení se může spustit měřicí proces. Během měření si program průběžně stahuje data z mikroprocesoru a zobrazuje průběžné výsledky. Po ukončení měřicího procesu se naměřená data zpracují a program navrhne nové impulsní konstanty pro kalibraci průtokoměrů. Z programu se dá ovládat také oběhové čerpadlo, které vhání vodu do průtokoměrné linky.

1. Měření průtoku

Průtok patří mezi nejčastěji měřené veličiny v průmyslu. Existují různé metody pro měření průtoku.

1.1 Objemový průtok

Objemový průtok Q_v vyjadřuje jaké objemové množství tekutiny dV proteče kanálem za čas dt . Využívá se pro nestlačitelné tekutiny.

$$Q_v = \frac{dV}{dt} \quad [\text{m}^3 \cdot \text{s}^{-1}]$$

Vzorec 1 - Výpočet objemového průtoku

1.2 Hmotnostní průtok

Hmotnostní průtok Q_m je definován jako hmotnostní množství tekutiny dm protečené kanálem za daný čas dt . Využívá se pro stlačitelné tekutiny.

$$Q_m = \frac{dm}{dt} \quad [\text{kg} \cdot \text{s}^{-1}]$$

Vzorec 2 - Výpočet hmotnostního průtoku

1.3 Měření v otevřeném kanálu

Otevřeným kanálem se myslí například koryto řeky, potok nebo odpadní stoka. Otevřený kanál může mít různé tvary. Profilu koryta se odborně říká přeliv. Přelivů je několik. Například: obdelníkový přeliv, lichoběžníkový přeliv, trojúhelníkový přeliv nebo kruhový přeliv.

Na otevřeném kanálu se dá měřit průtok hned několika způsoby. Jedním způsobem je sledovat výšku hladiny v čase, za předpokladu, že je známa šířka a tvar kanálu. Dále integrálem výšky hladiny jako funkce času a vynásobením koeficientem tvaru koryta a šířkou lze získat objem protečené kapaliny v daném čase.

Druhou metodou je sledování přepadu. U přepadu se sleduje změna rozdílu

výšky hladiny a výšky přepážky. Z těchto parametrů se zpočítá velikost průtoku.

Další metoda je měření průtoku pomocí hráze. Zde se měří určité množství natečené kapaliny do uzavřené přehrady za měřený čas.

1.4 Měření v uzavřeném kanálu

Uzavřeným kanálem se myslí například potrubí. V tomto druhu kanálů není možné měřit průtok například přepadem. V uzavřených kanálech se k měření průtoku používají průtokoměry. Průtokoměry se dělí na dvě kategorie pro měření hmotnostního průtoku a pro měření objemového průtoku.

Pro měření hmotnostního průtoku se využívají průtokoměry založené na měření Coriolisovy síly a kalometrické anemometry.

Coriolisovy průtokoměry obsahují dvě trubky, které se rozkmitají na rezonanční frekvenci. Jedna má konstantní fázi a ve druhé vzniká se změnou průtoku fázový posuv. Rozdílem těchto fází lze určit průtok. Velikost Coriolisovy síly závisí na pohybující se hmotě a její rychlosti.

Kalometrické anemometry mají do potrubí vyvedenou topnou spirálu. Tato spirála je ohřátá a protékající voda z ní odebírá teplo. Před spirálou je umístěno čidlo, které měří teplotu přitékající vody. Za spirálou je umístěno druhé čidlo. Čím je průtok pomalejší, tím více se voda ohřeje. Z rozdílu teplot čidel lze určit okamžitou rychlost proudění.

Pro objemové průtoky se používají plováčkové, lopatkové, rychlostní a dávkovací průtokoměry.

Plováčkový průtokoměr převádí rychlost kapaliny na polohu ručičky. V plováčkovém průtokoměru působí tlak protékající kapaliny na plováček (pružina vrací plováček), hýbe s ručičkou, která na stupnici ukazuje okamžitý průtok.

V *lopatkovém průtokoměru* způsobí protékající voda rotační pohyb lopatek. Energie z hřídele se přenesení na ručičku. Vodočet se přičítá spolu s otáčející se hřídelí, podobně jako funguje tachometr u automobilu. Tyto průtokoměry se využívají často v domácnostech jako vodoměry, měří bez přívodu elektrické energie.

Vírové senzory v sobě mají umístěnou překážku, clonku. Za touto překážkou

vznikají víry, ty překážku rozkmitají. Z naměřené frekvence se spočítá rychlost proudění. Používají se v průmyslu. Výhodou je rychlá přestavba, stačí pouze vyměnit clonku.

Ultrazvukové průtokoměry měří průtok na základě vyslání a příjmu ultrazvuku. Vysílač vyšle signál. Přijímač čeká na signál, když signál dorazí, vyhodnotí čas. Podle tohoto času lze určit rychlost průtoku. U ultrazvukového měření záleží na hustotě protékající kapaliny.

Indukční průtokoměry vytváření magnetické pole procházející potrubím v daném místě. V potrubí jsou umístěna čidla, na nichž se indukuje napětí úměrné okamžitému průtoku. Tento způsob měření je odolný proti nárazům, takže se s ním dá měřit velice přesně. Indukované napětí na čidlech je malé, takže se musí před zpracováním zesílit.

2. Kalibrace průtoků

2.1 Kalibrace

Kalibrace je soubor úkonů, kterými se za specifikovaných podmínek stanoví vztah mezi hodnotami veličin, které jsou indikovány měřicím systémem, měřicím přístrojem, hodnotami reprezentovanými ztělesněnou mírou nebo referenčním materiálem a odpovídajícími hodnotami, které jsou realizovány etalony (standards).

2.2 Kalibrace přes váhu

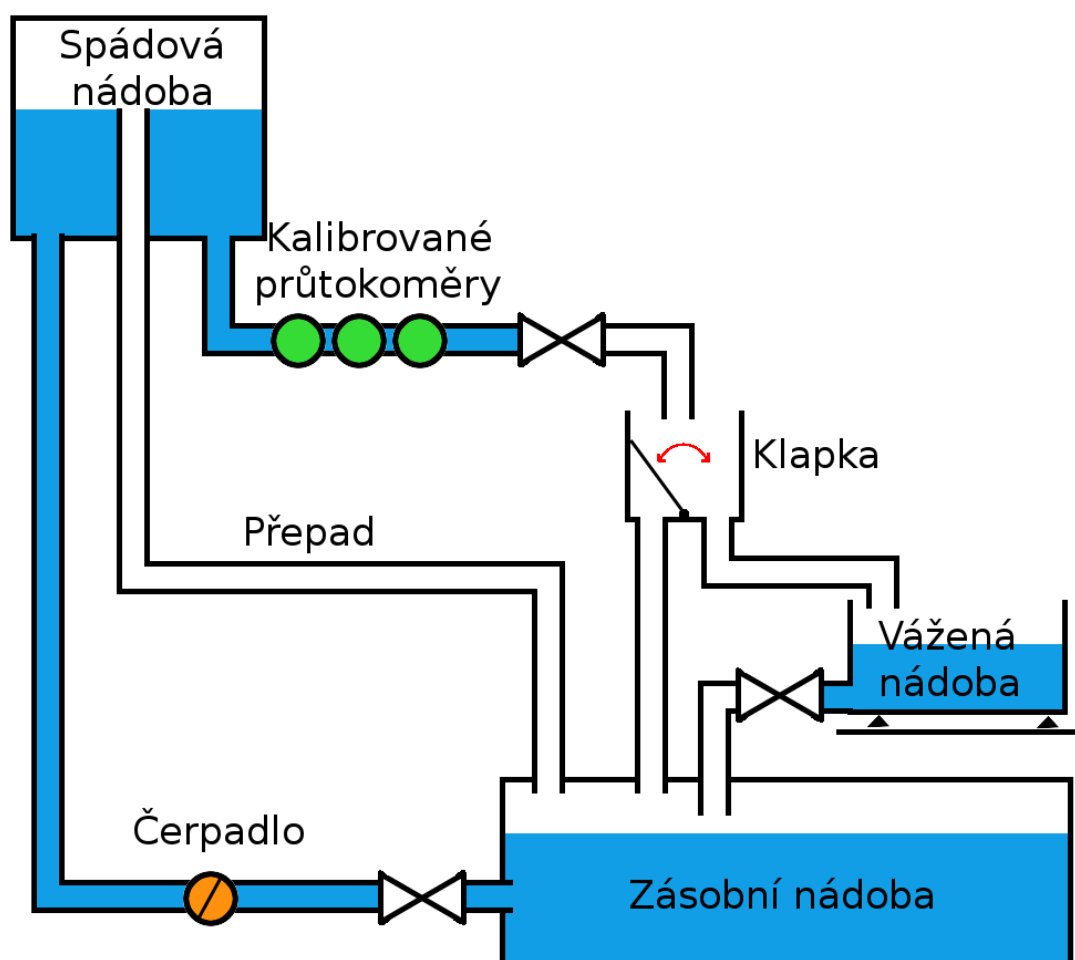
Kalibrace přes váhu se považuje za jednu z nejstabilnějších a nejpresnějších metod kalibrací průtoků. Tato metoda udává pouze střední hodnotu průtoku. Dělí se na metodu se statickým vážením a metodu s dynamickým vážením. U kalibrace objemových průtokoměrů je důležité udržovat při kalibraci konstantní tlak a konstantní teplotu. Jinak by mohlo dojít ke změně objemu měřené kapaliny a pak by mohla být do kalibrace vnesena kalibrační chyba.

2.2.1 Statické vážení

Princip statického vážení je založen na tom, že čerpadlo čerpá kapalinu ze zásobní nádoby do spádové. Ve spádové nádobě musí být udržována konstantní výška hladiny. Ta musí být konstantní, aby nedocházelo k poklesu tlaku v potrubí při poklesu hladiny. Tuto podmínku zajišťuje přepad.

Ze spádové nádoby vytéká samospádem kapalina. Ta protéká přes kalibrované průtokoměry a přes regulační ventil, kterým se reguluje tlak v potrubí. Z ventilu protéká kapalina přes klapku do vážené nádoby, která je umístěna na velice přesných vahách.

Spádová nádoba musí být zajištěna tak, aby se do měřené kapaliny nedostaly žádné nečistoty, které by mohly způsobit chybnou kalibraci.



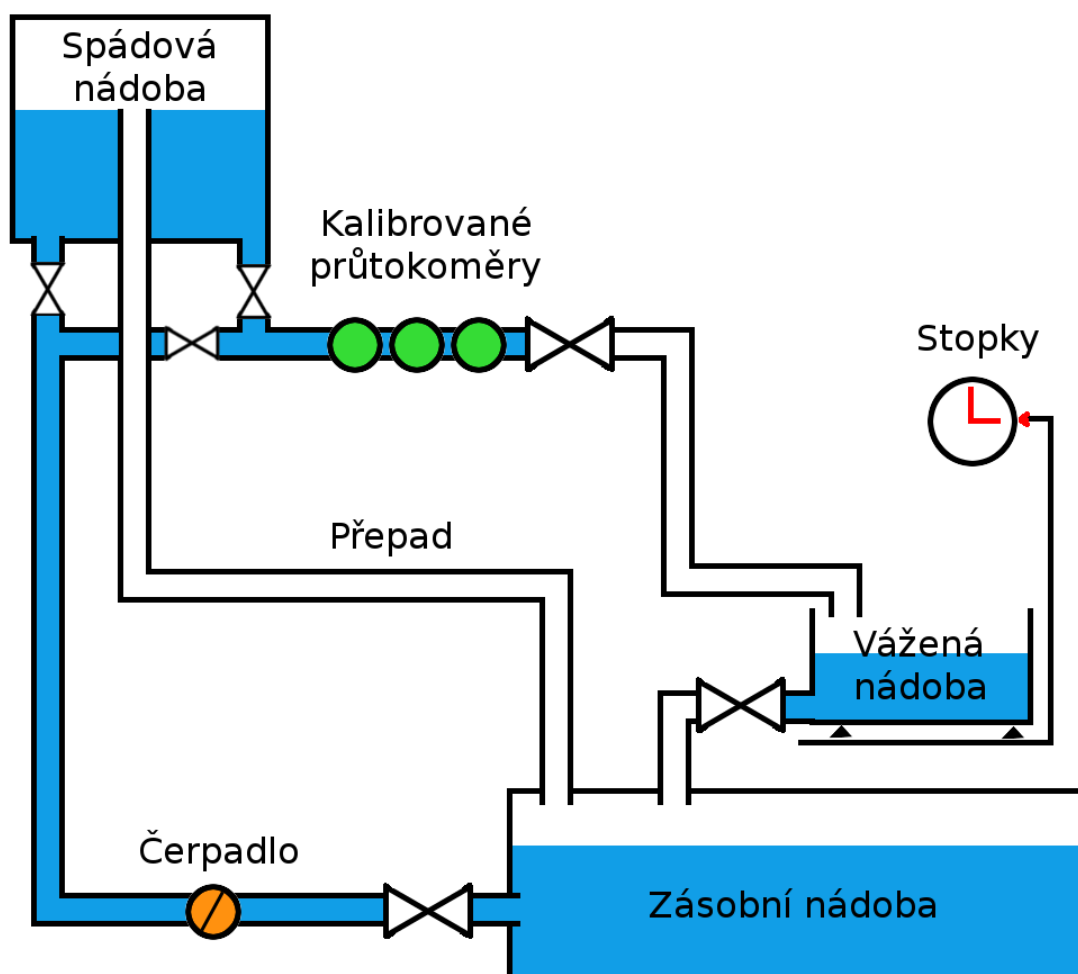
Obr. 1 – Metoda kalibrace statickým vážením

2.2.2 Dynamické vážení

Dynamické vážení je principiálně velice podobné jako statické vážení. Ze zásobní nádoby čerpá čerpadlo vodu do samospádové nádoby, kde je pomocí přepadu udržována konstantní výška hladiny. Odsud protéká samospádem voda přes kalibrované průtokoměry až do vážené nádoby.

Ve vážené nádobě jsou stanoveny dva hmotnostní body, podle kterých jsou ovládány stopky. Jeden o nižší hmotnosti a jeden o vyšší hmotnosti. Když dosáhne hmotnost natečené kapaliny nižšího hmotnostního bodu, spustí se stopky. Po dosažení vyššího hmotnostního bodu se stopky zastaví.

U této metody se mohou přesné váhy zaměnit přesnými hladinovými čidly, za předpokladu, že bude znám objem nádoby. Při nárůstu hladiny do výšky spodního čidla se stopky spustí, při nárůstu po vyšší čídlo se stopky zastaví.

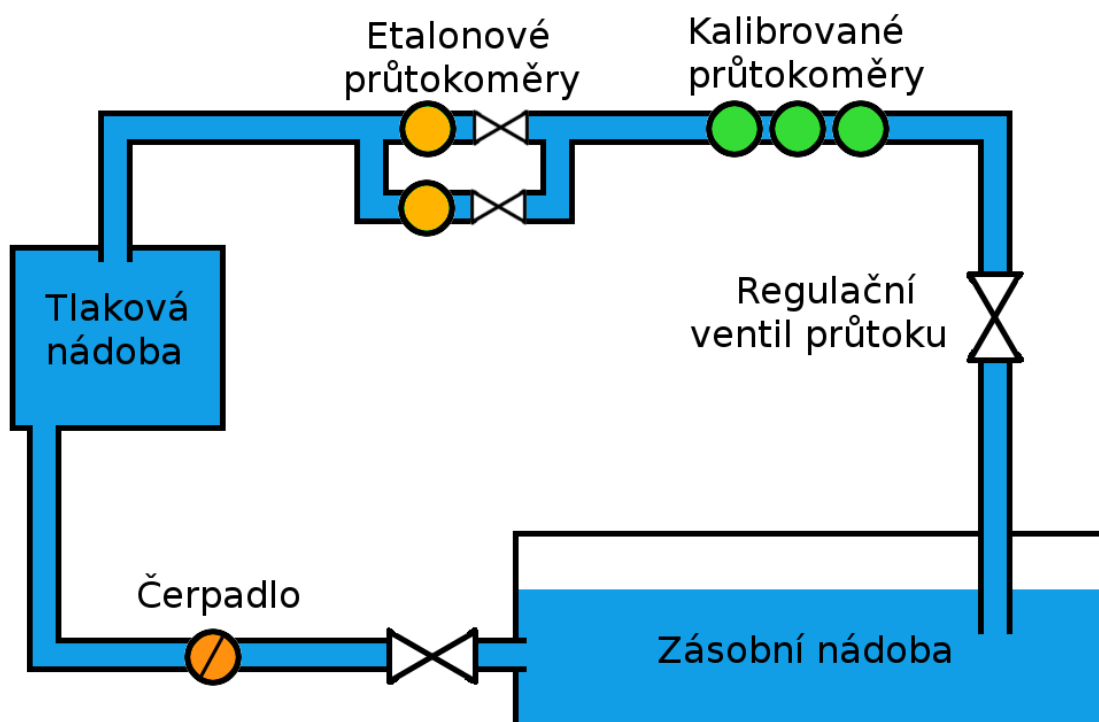


Obr. 2 – Metoda kalibrace dynamickým vážením

2.3 Oběhová kalibrace

Oběhová kalibrační metoda je založena na principu oběhu kapaliny v uzavřené smyčce potrubí.

Kapalina je čerpána ze zásobní nádoby a je posílána do tlakové nádoby. V tlakové nádobě se natlakuje potrubí na žádaný tlak. Odsud vede natlakovaná kapalina přes etalonové průtokoměry. Tyto průtokoměry se zpravidla zapojují paralelně a každý z nich je určen pro jiné průtoky. U příslušného etalonu se při používání otevře ventil, u ostatních se zavře. Dále protéká kapalina přes kalibrované průtokoměry a přes regulační ventil zpět do zásobovací nádoby. Regulační ventil slouží k nastavené tlaku v potrubí.



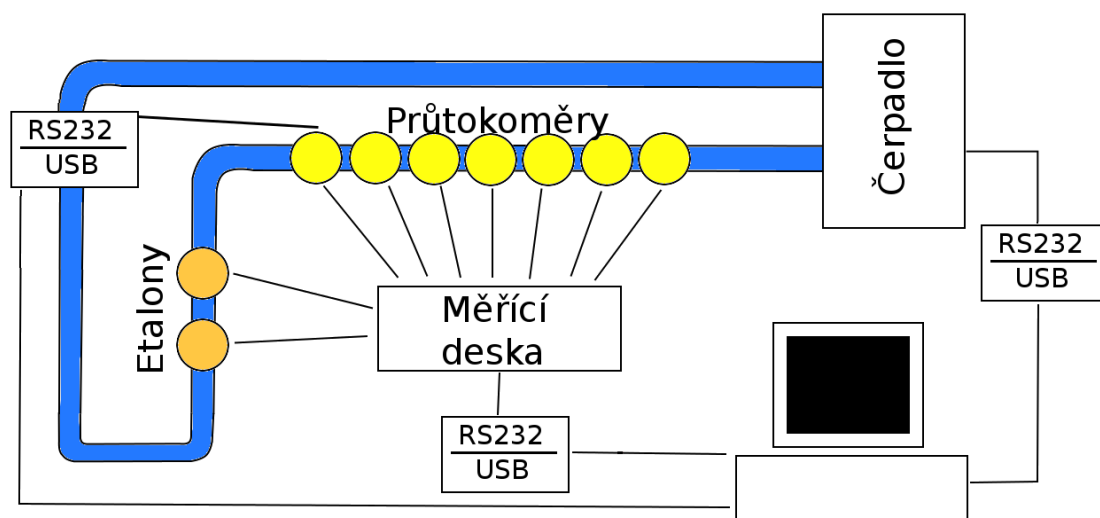
Obr. 3 – Metoda oběhové kalibrace

3. Praktická část

3.1 Kalibrační linka

Kalibrační linka, kterou se tato práce zabývá, je oběhová průtokoměrná linka, na níž může být umístěno až šestnáct průtokoměrných měřidel. Alespoň jedno z nich musí být již zkalibrovaný průtokoměr, podle kterého se kalibrují ostatní měřidla. Tento průtokoměr slouží jako etalon. Z toho vyplývá, že se na této kalibrační lince dá kalibrovat až patnáct průtokoměrů najednou.

Srdcem linky a zdrojem proudění vody v potrubí je oběhové čerpadlo, které čerpá tekutinu z nádrže a posílá ji do potrubí. Ta projde všemi průtokoměry, linkou a opět se vrací do nádrže. Průtokoměry jsou řazené za sebou. Tím je zajištěno, že bude ve všech měřidlech stejný okamžitý průtok. Na základě průtoku generují měřidla signály, impuls / objem kapaliny, nebo objem kapaliny / impuls, od 0 do 1 500 Hz podle okamžitého průtoku. Tyto signály jsou přivedeny na měřicí desku.



Obr. 4 – Kalibrační linka - blokové schéma

Na měřicí desce jsou umístěny výměnné moduly, které mají za úkol galvanicky oddělit signály generované průtokoměry od mikroprocesoru a provést změnu úrovně signálu na amplitudu 0 až 3,3 V, což odpovídá logice CMOS. Diskrétní signály jsou přivedeny na porty mikroprocesoru. Procesor tyto porty skenuje s určitou

vzorkovací frekvencí. Po naskenování provádí detekci náběžných hran. Po detekci změny náběžných hran dojde k zaznamenání příchozího pulsu příslušnému průtokoměru. Po spuštění měřicího procesu se začne měřit čas. S příchodem prvního pulsu se čas zaznamená. Podobně jako se měří čas od spuštění k prvnímu pulsu měří se i čas od ukončení procesu k prvnímu příchozímu pulsu. Tyto časové hodnoty se využívají pro zpřesnění měření na desetiny pulsů. Tento způsob měření usnadní a zpřesní výpočet kalibračních konstant. Zejména u kalibrace na malých průtocích se měřicí proces značně urychlí. Průběžné a konečné výsledky jsou poskytovány obslužnému programu na vyžádání.

Počítačový program řídí měřicí proces na desce. V programu je implementován časovač s funkcí odpočtu, kterému se nastaví délka měřicího cyklu. Společně se spuštěním časovače vynutí program spuštění měřicího procesu v desce. Po vypršení časovače se vynutí ukončení měřicího procesu. Společně s ukončením si vyžádá aktualizaci naměřených dat. Ta jsou aktualizována i v průběhu měření. Po dokončení měřicího procesu jsou z nich vypočítány kalibrační konstanty.

Pro komunikaci mezi počítačem a deskou je použit tzv. universální asynchronní vysílač/přijímač, neboli sériový port. Pro zajištění co největší věrnosti přenosu dat je využit jednoduchý přenosový protokol. Součástí protokolu je i kontrolní součet CRC.

Do průtokoměrů jsou pomocí stejného rozhraní, jako u měřicí desky, nahrány kalibrační konstanty.

Pomocí obslužného programu lze nastavit i průtok čerpadla. Program nahraje do mikroprocesoru číslíkovou hodnotu. Procesor pomocí D/A převodníku vytvoří analogovou hodnotu napětí o velikosti 0 až 10V a čerpadlo nastaví požadovaný průtok.

3.2 Měřicí deska

Pro sběr dat byla vyvinuta měřicí deska (měřicí ústředna). Hlavním úkolem měřicí desky je zpracování signálů, sběr a zaznamenávání dat získaných z průtokoměrů.

Na desku jsou přivedeny signály od průtokoměrů. Mají obdelníkový charakter. Pro další zpracování je nutné převést analogový signál na diskrétní. Pro tuto operaci jsou na desce umístěny vyměnitelné moduly, které pomocí diod a optočlenu transformují signály obdelníkového průběhu o různé amplitudě na stejný signál o amplitudě 0 až 3,3 V. Rozmezí amplitud odpovídá logice CMOS. Výhodou těchto flexibilních modulů je zajištění galvanického oddělení měřicí desky od měřicí soustavy. Moduly se dají libovolně zaměňovat, takže při vzniklé poruše se poškozený modul jednoduše vymění za jiný. Výstupní obdelníkové signály jsou přivedeny na vstupněvýstupní porty mikroprocesoru.

Pro tuto aplikaci byl použit mikroprocesor řady 8051 firmy Silabs s typovým označením c8051f020. Programuje se přes rozhraní JTAG. Je taktován krystalem o kmitočtu 22,1184 MHz. Mikroprocesor plní tři hlavní funkce, kterými je Inicializace komponent, sériová komunikace a měřicí program.

Inicializace komponent se provádí vždy po připojení měřicí desky k napájecímu napětí, nebo po každém stisknutí tlačítka reset. Jejím úkolem je implementovat komponenty, které se používají v programu a nastavit jim požadované hodnoty. Jako první věc, která se provede, je zakázání tzv. Watch dogu. Dalším úkolem je zakázání vnitřního oscilátoru mikroprocesoru a nastavení na taktování externím krystalem o frekvenci mnohem vyšší taktovací frekvencí. Následuje nastavení CrossBaru.

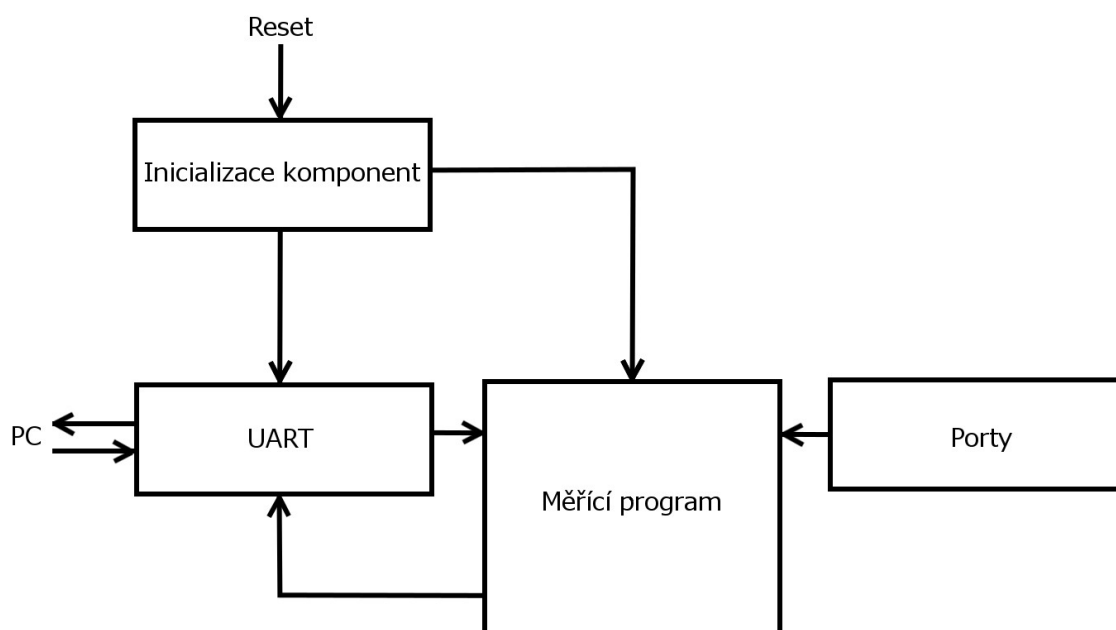
CrossBarr je nástroj, který na piny portů P0 až P3 routuje použité periferie dle předem dané posloupnosti. Crossbar se nastavuje podle tabulky uvedené v datasheetu mikroprocesoru. Pomocí tabulky byly na port P0 naroutovány UART0, SPI0, CEX0 a /INT0. Nastavení se provádí nahráním konkrétních hodnot do bytů XBR0, XBR1 a XBR2.

Po nastavení Crossbaru přijde inicializace periférií. Jako první z periférií

se inicializují čítače/časovače. K jejich nastavení se využívá bytu s názvem TMOD, ve kterém se dají nastavit vlastnosti, přičemž horní čtyři bity slouží k nastavení vlastností čítače/časovače1 a spodní čtyři bity k nastavení čítače/časovače0. Primární funkcí čítače/časovače0 je vytváření vzorkovací frekvence pro skenování portů. T0 je nastaven jako časovač s přednastavením. Čítač/časovač1 je nastaven tudíž jako časovač s přednastavením. Je využíván jako synchronizační hodiny pro sériový přenos.

Společně s čítači/časovači se inicializuje také sériový kanál UART. Inicializací se přednastaví základní parametry pro přenos. Je nastaven jako osmibitové asynchronní rozhraní. K jeho nastavení slouží registr SMOD.

Jako poslední je nutné nastavit povolení přerušení. K nastavení slouží registr IE, interrupt enable. Dá se v něm nastavit několik druhů přerušení, jako je přerušení od sériového kanálu, časovačů, vnější přerušení a podobně. Společně s nastavením konkrétního přerušení je nutné nastavit nejvyšší bit registru, s označením EA, protože jinak jsou všechna přerušení zakázána. Přerušení nastane, vždy když přeteče časovač nebo sériový kanál obdrží nový znak. V takové situaci se nastaví v registru IP příslušný bit, který signalizuje, že nastalo přerušení a nazývá se příznak přerušení. Při přerušení se vykoná daný podprogram a poté se automaticky vynuluje příznak přerušení. Při nevyužití podprogramu přerušení se musí příznak přerušení vynulovat programově.



Obr. 5 – Program měřicí desky – blokové schéma

Druhým blokem instrukcí programu mikroprocesoru se provádí společně s přerušením sérového kanálu. Hlavní činností tohoto bloku je komunikace s obslužným programem počítače, který měřicí desku ovládá. Program vyšle z počítače informaci o tom, co se má provést. Sériový kanál přijme informaci, která se vyhodnotí, a pak provede danou operaci. Tou může být spuštění či zastavení měření, nebo odeslání naměřených dat.

Měřicí program skenuje porty se vzorkovací frekvencí, kterou udává časovač T_0 . Pro splnění vzorkovacího teorému musí být vzorkovací frekvence alespoň dvakrát větší, než je maximální možná frekvence vzorkovaného signálu. Vzorkovací frekvence je přibližně desetkrát větší, než je maximální dosažitelná frekvence signálů generovaných průtokoměry. Po skenování jsou ze vzorků detekovány náběžné hrany. Po získání vektoru náběžných hran je každému z průtokoměrů, kterému byla detekována náběžná hrana, pomyslně přiřčen puls. Jeden puls odpovídá délce jedné periody.

Společně s čítáním celých pulsů se měří doba zpoždění náběžné hrany vstupního signálu od spuštění až po ukončení měřicího procesu. Doba se měří od spuštění měření k první náběžné hraně. Obdobné měření probíhá také po ukončení měřicího procesu. Doba se měří od poslední náběžné hrany k ukončení měření. Z těchto dvou údajů se určuje doba zpoždění signálu, díky kterému se dá určit délka necelé periody. Těchto necelých period se využívá zejména při kalibraci nízkých průtoků. Poslední měřenou veličinou je počet vzorkovacích period časovače mezi dvěmi náběžnými hranami.

3.2.1 Čítače a časovače

Čítače a časovače jsou základními periferiemi všech mikroprocesorů. Jsou určeny pro měření času, frekvencí, nebo čítání pulsů. Ve skutečnosti se jedná o periferie, které reagují na změnu logické hodnoty vstupního signálu. Rozdíl mezi čítačem a časovačem je pouze ve volbě zdroje vstupního signálu. Zdrojem může být buď vnitřní oscilátor mikroprocesoru, nebo změna vnější události. Dalo by se tedy říci, že časovač je obyčejný čítač hodinových pulsů. Při změně logické hodnoty vstupního signálu dojde k inkrementaci registru TL_n , nebo TH_n dle nastavení. Po přetečení těchto registrů je možné vykonat podprogram přerušení.

Před použitím čítačů/časovačů je nutné provést jejich vnitřní nastavení. To spočívá v nastavením několika základních parametrů v osmibitovém registru TMOD, který je rozdělen na horní čtyři bity pro nastavení čítače/časovače1 a spodní čtyři bity pro nastavení čítače/časovače0.

Prvním parametrem, který je třeba nastavit je GATE, neboli řízení hradlování. To rozhoduje o způsobu aktivace čítače/časovače. Je-li GATE = 0, je možné provést aktivaci pouze programově. Programová aktivace se provádí nastavením bitu TRn z bytu TCON log. 1. Je-li GATE = 1 aktivace se provádí jak programově, tak i obvodově. Obvodová aktivace se provede po přivedení logické hodnoty 1 na vstupní pin s označením INTn. V tomto případě je proveden logický součet OR mezi TRn a INTn, takže pokud nejsou obě dvě vstupní hodnoty nulové, časovač je aktivní.



Obr. 6 – Registr TMOD

Další parametr rozhoduje o tom, zda se jedná o čítač, nebo časovač. Tímto parametrem je bit z registru TMOD s označením C/T. Při jeho nastavení do log.0 je čítač/časovač nastaven jako časovač. Jako zdroj impulsů je nyní nastaven vnitřní oscilátor mikroprocesoru. Časovač čítá s frekvencí jednoho strojového cyklu, což je dvanáctina frekvence vnějšího krystalu. Tento krystal udává takt procesoru. Nastavení C/T do log.1 se změní čítač/časovač na čítač vnějších událostí. V tomto případě se stane zdrojem impulsů pin Tn. Při změně signálu na tomto pinu z log.1 na log.0, tedy při sestupné hraně, dojde k inkrementaci vnitřních registrů THn, nebo TLn.

Čítač/časovač lze nastavit do jednoho ze čtyř pracovních módů. Ty se nastavují pomocí bitů M1 a M0 v registru TMOD. V módu nula, M0 = 0 a M1 = 0, je nastaven jako osmibitový čítač s pětibitovým předděličem. Chová se stejně jako třináctibitový čítač, přičemž spodních 8bitů čítače tvoří registr TLn a horních 5 bitů registr THn. Když je hodnota THn a TLn rovna 2^{13} dojde k přetečení registru, THn a TLn se nastaví

do nuly. V registru TCON se zároveň nastaví příznak přetečení TF a vyvolá se přerušení od čítače/časovače, pokud je povoleno.

Mód 1 je stejný jako předchozí mód s tím rozdílem, že čítač/časovač je šestnáctibitový. Když načítá až do hodnoty 2^{16} , dojde k přetečení. Registry se nastaví na nulové hodnoty a vykoná se podprogram přerušení, je-li přerušení povoleno.

V módu dva je čítač/časovač nastaven jako osmibitový čítač s přednastavením. Před spuštěním je možné zvolit si hodnoty THn a TLn. Toto přednastavení je možné i v předchozích módech. Po spuštění čítače/časovače se TLn začne inkrementovat od přednastavené hodnoty až do hodnoty 2^8 , poté dojde k přetečení. Společně s přetečením se TLn automaticky nastaví hodnotu rovnou THn, která slouží jako konstanta. Tohoto módu se využívá zejména u generování cyklicky se opakujících časových pulsů, jejichž délku si lze zvolit. Doba jednoho ticku časovače se dá vypočítat z hodnoty frekvence použitého krystalu. Za předpokladu, že jeden strojový cyklus trvá 12 period oscilátoru, bude jeden tick časovače trvat $F/12$.

Poslední je mód dvou osmibitových čítačů. Tento mód využívá pouze čítače/časovače0. Jeden čítač pracuje s TL0 a může být aktivován jak obvodově, tak i programově, pomocí TR0. Druhý čítač pracuje s TH0 a aktivuje se pouze programově pomocí TR1. V tomto módu se může využívat současně časovač T1 pouze pro generování synchronizaci sériového kanálu. Nelze využít jeho přerušení, to využívá jeden z čítačů T0.

3.2.2 Sériový port

Sériový port je prostředek, který slouží k přenosu dat mezi měřicí deskou a počítačem. Jeho výhodou je, že při přenosu dat nevznikají kolize. Pro přenos stačí použít třívodičový kabel s vodiči RxD, TxD a GND. RxD je vodič, který slouží pro příjem. TxD je vodič, který slouží pro vysílání. GND je zemnicí vodič propojující země na vysílací straně a přijímací straně. Země je nutno propojit kvůli možné identifikaci úrovní signálu. Vodiče TxD a RxD se ve vodiči musí překřížit tak, aby TxD na jednom konektoru kabelu bylo přivedeno na RxD druhého konektoru. Pokud není splněna tato podmínka, obě dvě strany budou vysílat do stejného kabelu, tudíž ani jedna strana nic neuslyší.

Pro přenos dat sériovým kanálem je třeba nastavit základní parametry, jakými jsou přenosová rychlost a mód, ve kterém sériový kanál pracuje. Může pracovat ve čtyřech módech. Buď jako osmibitový synchronní přenos bez rámcové synchronizace, jako devítibitový asynchronní přenos s generováním přenosové rychlosti oscilátorem mikroprocesoru, nebo jako osmibitový, či devítibitový asynchronní přenos s přenosovou modulační rychlostí generovanou časovačem T1.

Pro tuto aplikaci byl zvolen osmibitový asynchronní přenos dat. Sériový kanál přenáší osm bitů, jeden byte, který je doplněn o start bit a stop bit. Start bit se vysílá vždy před daty a má hodnotu log. 0. Stop bit se vysílá vždy po odvysílání dat a jeho hodnota je vždy log. 1. Tyto bity slouží k synchronizaci odesílatele a příjemce. V případě, že příjemce detekuje start bit jako log. 1, nebo stop bit jako log. 0, došlo při přenosu k chybě.

Pro správnost přenosu je důležité zvolit na obou stranách komunikační smyčky stejnou přenosovou rychlost. Hodnota přenosové rychlosti pro sériovou komunikaci byla zvolena 115 200 Baud. Tuto rychlost vytváří časovač1, který je nastaven v módu 2, jako osmibitový časovač s přednastavením. Když dojde k přetečení časovače, do TH1 se nastaví hodnota TH1, která slouží jako konstanta a odpovídá přenosové rychlosti. Přenosová rychlost se spočítá z frekvence oscilátoru, která se rovná frekvenci použitého krystalu.

$$\text{přenosová rychlost} = \frac{2^{SMOD}}{32} \cdot \frac{f_{OSC}}{12 \cdot (256 - TH1)}$$

Vzorec 3 – Výpočet modulační rychlosti RS232

Sériový port se chová tak, že při odesílání a přijímání dat nastaví bit RI respektive TI do log. 1, což vyvolá přerušení od sériového kanálu. V přerušení se musí rozlišit, zda se jedná o přijímání, nebo odesílání dat. Příznak přerušení od sériového kanálu se po vykonání podprogramu nezruší, musí se vynulovat programově. Při příjmu dat se plní registr SBUF po bytech. Byte z tohoto registru se musí zpracovat dříve, než přijde nový. Pokud přijde nový byte dřív, než je SBUF vyčten, data v SBUF se přepíší. Pro odeslání stačí nahrát do registru SBUF žádaná data postupně po bytech a ta se postupně odešlou.

Komunikace s počítačem je řízena pomocí přerušení od sériového kanálu. Počítač řídí mikroprocesor pomocí bytů a ten řídí procesy na měřicí desce. Přijatá data se ukládají do registru SBUF0, z kterého se vyčítají do proměnné s názvem Byte. Po vyčtení bufferu začne procesor analyzovat přijatá data. Podle přijatých dat rozhodne, která operace se má provést. Má na výběr ze tří možností. Spustí měření, ukončí měření, nebo odešle žádaná data. Měření může spustit, pokud není právě spuštěno a pokud probíhá, může ho ukončit. Počítač si opakovaně může vyžádat data. Data jsou uložena jako šestnáctibitový unsigned integer. Před odesláním jsou data vložena do funkce, která je připraví pro odeslání. Tato funkce rozdělí každý integer po osmi bitech na vyšší a nižší byte. Tato dva byty jsou v počítači opět složeny do původního tvaru.

Při sériové komunikaci mohou vznikat poruchy přenosu, proto není výhodné posílat surová data. Pro zajištění co největší věrnosti byl vytvořen jednoduchý protokol, do kterého se data vkládají. K naplnění protokolu data se využívá funkce, která vytvoří a připraví data pro odeslání podle odesílacího protokolu.



obr. 7 – Odesílací protokol

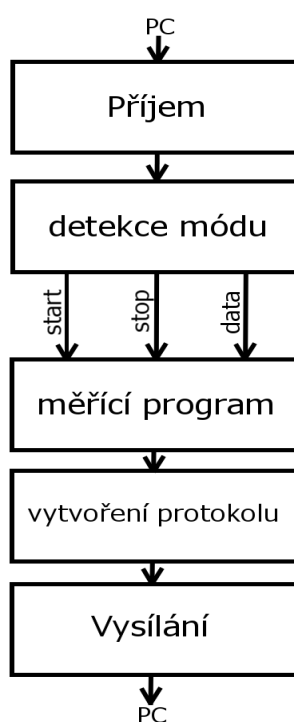
Naměřená data jsou uložena jako pole 16 bitových bezznaménkových integerů. Tato čísla jsou rozdělena vždy na nižších 8 bitů a vyšších 8 bitů, čili na 2 byty. Tato funkce uspořádá 32 datových bytů, v pořadí nižší byte a vyšší byte, které se doplní o dva informační byty na začátku a jeden na konci odesílacího cyklu. První byte protokolu určuje celkový počet bytů, které budou vyslány. Druhý byte určuje o jaký druh přenášených dat se jedná. Poslední datový byte je kontrolní součet, neboli CRC.

Byte se jménem *počet vyslaných bytů* informuje příjemce o tom, kolik má očekávat příchozích bytů. Hodnota se rovná celkovému počtu datových bytů a třem bytům informativním. Byte *mode* informuje příjemce o tom, jaká data jsou přenášena. Zda se jedná o celkové pulsy, zpoždění signálu, nebo délku periody signálu.

Byte *CRC*, neboli kontrolní součet, slouží příjemci k ověření věrohodnosti dat.

Když příjemce provede stejný algoritmus jako odesílatel, získá kontrolní součet. Když si jsou kontrolní součty rovny, přijatá data jsou věrohodná. Kontrolní logický součet vznikne exkluzivním logickým součtem, funkcí XOR, všech předchozích vyslaných bytů.

Všechny byty jsou průběžně vkládány do vysílacího bufferu SBUF0 následně odeslány přes sériovou linku do obslužného programu v počítači.



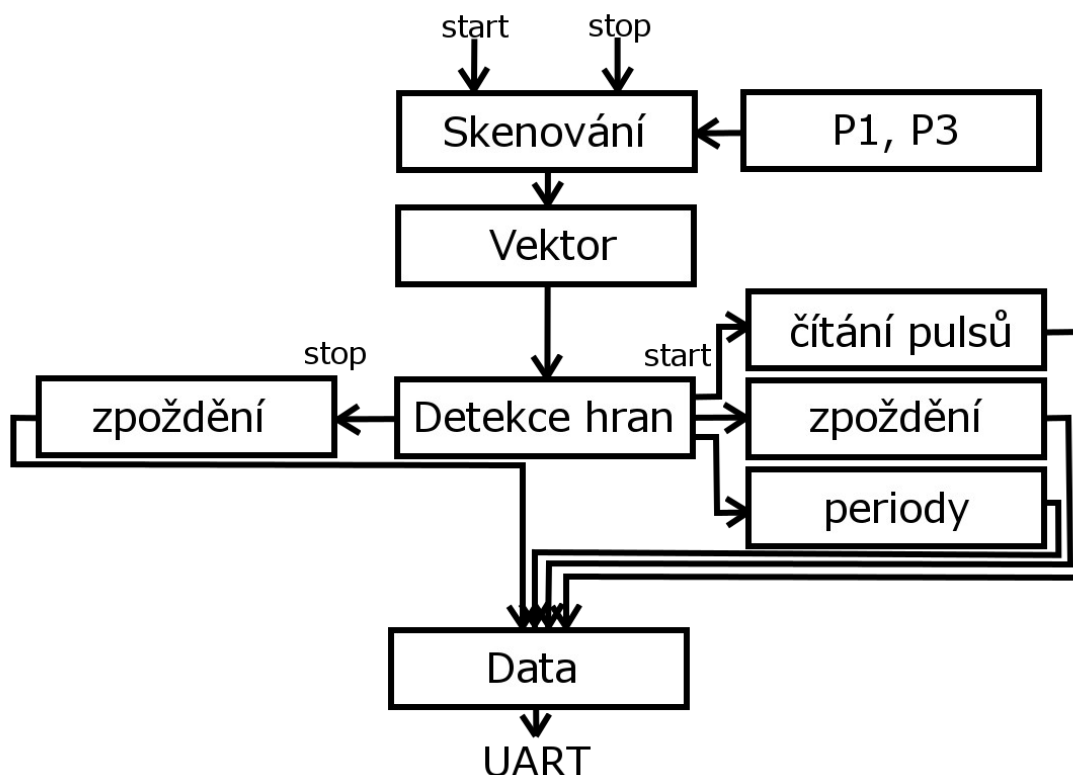
obr. 8 – Funkce přerušení UART – blokové schéma

3.2.3 Měřicí program

Hlavním úkolem mikroprocesoru je vykonávání měřicího programu. Program má za úkol sbírat a vyhodnocovat data získaná z impulsních výstupů průtokoměrných měřidel umístěných na kalibrační lince. Signály z průtokoměrů jsou upraveny pro zpracování mikroprocesorem pomocí modulů na desce, přes které signály prochází. Měřené průtokoměry generují sinusový signál o frekvenci 0 až 1 500 Hz, přičemž

okamžitý průtok je úměrný okamžitému kmitočtu. Mezi měřenými průtokoměry musí být vždy umístěn alespoň jeden, který byl již zkalibrován. Vůči němu se poté porovnává věrohodnost naměřených dat. Program je napsán v jazyce C.

Měřicí program je řízen z přerušení sériového kanálu, který dostává instrukce od obslužného počítačového programu. Po obdržení instrukce zpracovává a řídí další procesy. Program je ovládán instrukcemi *start* a *stop*.



Obr. 9 – Měřicí program – blokové schéma

Po přijetí instrukce *start* se měřicí program nastaví do měřicího módu *start*. Současně se provede inicializace a nastavení proměnných a registrů, které se při měření využívají. K měření se využívá časovač mikroprocesoru Timer0. Měřicí proces se spouští jeho aktivací. Po vynucení spuštění programu se časovač spustí. Časovač je nastaven v módu osmibitového časovače s přednastavením. Je přednastaven tak, že jeho přerušení nastává s určitou frekvencí, čehož se využívá pro vzorkování měřených signálů. Pro vzorkování signálů musí být dodržen Shannon-Kotelnikovův

teorém. Vzorkovací frekvence musí být alespoň dvojnásobná, než frekvence vzorkovaná. Při nedodržení této podmínky dochází k aliasingu a získaná informace je mylná.

Když nastane přerušení od časovače, dojde k *naskenování* portů P1 a P3, na které jsou přivedeny signály z průtokoměrů převedené na signál 0 až 3,3V. Na portech procesoru jsou buď log.1, nebo log.0. Jelikož mikroprocesor i signály pracují v negativní logice, jsou hodnoty portů znegované, kvůli práci s pozitivní logikou. Pro získání správných hodnot nemusí být vstupní porty negovány. Pro zjednodušení programu se z těchto dvou bytů vytvoří jeden vektor vstupních hodnot 16 bitový unsigned integer. Byte získaný z portu P3 se uloží do vektoru na horních osm bitů a byte získaný z portu P1 na spodních osm bitů. Pro vytvoření tohoto vektoru slouží funkce, která vloží do vektoru byte z portu P3 a celý vektor posune o osm míst doleva, což je ekvivalentní operace, jako vynásobit vektor číslem 2^8 . V tuto chvíli je horních osm bitů obsazených bytem získaným z portu P3 a spodních osm bitů je rovno nule. Pro dokončení funkce se k hodnotě vektoru přičte byte získaný z P1. V tuto chvíli obsahuje vektor vstupů šestnáct bitů. Každý bit odpovídá navzorkované hodnotě signálu příslušející konkrétnímu průtokoměru v sestupném pořadí, MSB signalizuje průtokoměr 16 a LSB průtokoměr 1.

Pro čítání pulsů je výhodné sledovat buď náběžné, nebo sestupné hrany signálů. Zde jsou sledovány náběžné hrany. Každé zaznamenání náběžné hrany signalizuje příchozí puls. Každý příchozí puls by měl být přičten příslušnému průtokoměru. Pro detekci náběžných hran je použita funkce, která pracuje se současnou i minulou hodnotou vektoru vstupních signálů. Pomocí logické bitové operace XOR, exkluzivního logického součtu, se dají detekovat změny hran. Tato operace porovnává dvě čísla po bitech, pokud mají dva porovnávané bity stejnou hodnotu výsledek je 0, pokud se liší výsledek je 1. Aplikováním funkce XOR na minulé a současné hodnoty vektoru získá procesor vektor, který signalizuje změny hran. Náběžná hrana se vyznačuje tím, že dvě po sobě jdoucí hodnoty diskrétního signálu mají logické hodnoty [0, 1]. To znamená, že když je příslušný bit vektoru minulých hodnot roven 0 a současný bit téhož vektoru roven 1, jedná se o náběžnou hranu. V opačném případě jde o hranu sestupnou. Pro určení náběžné hrany z vektoru, který signalizuje všechny hrany, stačí provést s vektorem logickou bitovou operaci AND, s hodnotou vektoru vstupů současných

hodnot. Protože když je současná hodnota vstupu v logické 1, tak buď ke změně hrany nedošlo, potom je příslušný bit ve vektoru hran roven nule a po provedení logického součtu bude výsledek roven nule, nebo došlo ke změně hrany a po logickém součtu bude výsledek roven jedné, což signalizuje změnu signálu náběžnou hranou. Ve skutečnosti se tyto logické operace provádějí s celými šestnáctibitovými čísly. Výsledkem této funkce je vektor náběžných hran, který se mění se stejnou vzorkovací frekvencí, jakou se vzorkují vstupní signály.

Až dosud při vykonávání programu nezáleželo, zda počítač vyslal požadavek start nebo stop. Rozdíl mezi režimem *start* a *stop* spočívá v tom, že režim *start* probíhá stále dokola, dokud nepřijde požadavek *stop*, poté se ukončí. Jeho úkolem je změřit zpoždění od spuštění časovače po příchod prvního pulsu, čítání pulsů a změření délky periody. Režim *stop* se začne vykonávat po ukončení režimu *start*. Jeho úkolem je změřit dobu od poslední hrany konkrétního signálu po okamžik, kdy byl přijat požadavek *stop*.

Hlavním úkolem režimu *start*, nebo-li měřicího režimu, je čítání příchozích pulsů o různých frekvencích po určitou dobu. Pro detekci příchozích pulsů se porovnává vektor náběžných hran. Pokud má větší hodnotu než nulovou, signalizuje to příchozí pulsy. V případě, že je vektor nulový, je zbytečné provádět další funkce, protože počet příchozích pulsů je nulový. V případě, že přišel alespoň jeden puls, je nutné ho zaznamenat. Jestliže není nulový, je nutné provést několik operací pro přiřazení pulsu správnému průtokoměru. Ve vektoru může být zaznamenán jeden puls, ale může jich být zaznamenáno až šestnáct najednou. Proto je potřeba porovnat bity vektoru a rozhodnout, od kterých průtokoměrů pulsy přišly. Pro toto určení se využívá *for* cyklu, který vždy vykoná šestnáct cyklů. V každém cyklu se porovnává nejnižší bit vektoru. Pokud je větší než nula, příchozí puls patří průtokoměru s číslem, které odpovídá číslu cyklu a provede se jeho zaznamenání. V opačném případě se neprovede žádná operace a program pokračuje. Před začátkem dalšího cyklu se provede rotace vektoru doprava o jedno místo. To znamená, že bit s nejnižší váhou se přenesse na pozici bitu s nejvyšší váhou a ostatní bity se posunou o jedno místo. Po provedení všech šestnácti cyklů se opět dostanou bity na původní místa. Výhodou použití *for* cyklu je, že se otočí pořadí průtokoměrů od nejnižšího po nejvyšší číslo. Pro zaznamenávání pulsů bylo inicializováno pole, které se skládá z šestnácti šestnáctibitových bezznaménkových

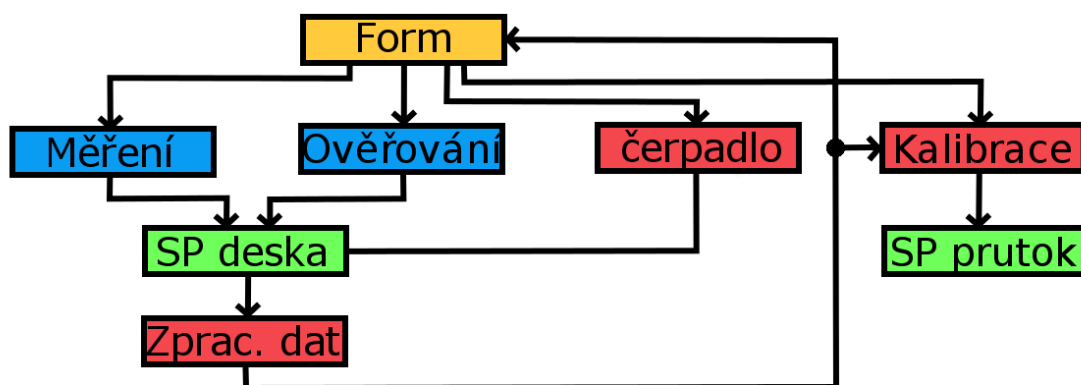
proměnných hodnot. Puls se zaznamenává inkrementací příslušné proměnné v poli. Stejně pole bylo vytvořeno pro hodnotu zpoždění a délku periody signálů.

Spolu s detekcí hran se zaznamenává také zpoždění mezi spuštěním měřicího procesu a prvním pulsem. Časovač generující vzorkovací frekvenci společně se skenováním vstupních portů také zaznamenává, kolikrát přerušení nastalo, tedy kolik proběhlo vzorkovacích period. S příchozím pulsem se zjistí, jestli je hodnota zpoždění konkrétního průtokoměru nulová, pokud ano, tak se do ní запиše počet vzorkovacích period. Tato operace se provede pouze při příchodu prvního pulsu. Podobně jako doba zpoždění se určuje i délka periody signálu. Délka periody je počet vzorkovacích period mezi dvěma pulsy. Podle teoretického výpočtu délky periody časovače lze vypočítat frekvenci měřených signálů. Délka periody je důležitá pro výpočet necelé periody, o kterou je signál zpožděn. Toto číslo je rovno podílu délky zpoždění a délky periody.

Hlavní úlohou režimu *stop* je určit délku necelé periody, která vznikne mezi posledním zaznamenaným pulsem a místem, ve kterém byl přijat požadavek ukončení měřicího procesu. Po přijetí požadavku *stop* se nastaví měřicí program do režimu *stop* a přestane čítat příchozí pulsy. Zaznamenává je však stále. Ve chvíli, kdy se změní režim *start* na režim *stop* se vynuluje čítač, který čítá proběhlé periody vzorkovací frekvence a začne čítat od nuly. Stejně jako v režimu *start* se pomocí *for cyklu* měří jaké je zpoždění od požadavku na ukončení měření po první přijatý puls každého průtokoměru. K získání délky necelé periody signálu, od posledního pulsu po ukončení měření, stačí od délky celé periody odečíst naměřené zpoždění. Tato konečná necelá perioda se přičte k té, která se určila na začátku měření. Po ukončení tohoto procesu jsou data poskytnuta pro odeslání.

3.3 Obslužný program

Pro ovládání měřicí desky, řízení čerpadla a kalibraci průtokoměrů byl zkonstruován počítačový program. Je napsán v programovacím jazyce C#. Byl vyvíjen v prostředí Microsoft Visual Studio Express 2010, což je volná verze Visual Studia. Je to GUI aplikace. Slouží jako uživatelské rozhraní mezi uživatelem a systémem. Program se skládá z několika tříd. Každá třída se zabývá konkrétním úkolem.



Obr. 10 - Obslužný program – blokové schéma

Po spuštění programu vyplní uživatel informace o kalibrovaných a kalibračních měřidlech. Poté nastaví průtok média čerpadlem a limit měření. Po nastavení může být spuštěno měření. Měření probíhá do vypršení časového limitu. Během procesu se v tabulce na záložce průtoky zobrazují okamžité průtoky. Je zde zobrazena i historie průtoků. Současně se v záložce kalibrace zobrazují celkové počty vygenerovaných pulsů průtokoměry a jejich difference mezi kalibračním a měřenými průtokoměry. Po vypršení limitu je měřicí proces ukončen. Po stažení původních kalibračních konstant z kalibrovaných průtokoměrů jsou na základě naměřených dat dopočítány nové kalibrační konstanty, kterými mohou být průtokoměry zkalibrovány. Pro zkalibrování se přístroje otestují na třech různých průtocích. Po otestování je vystaven kalibrační protokol s informacemi o kalibraci.

3.3.1 Grafické rozhraní

Třída, udává vzhled a podobu celého programu se nazývá *Form*. Vytváří grafické rozhraní mezi uživatelem a kalibračním systémem. Jsou v něm implementovány různé objekty jako například textboxy, buttony či richtextboxy, pomocí nichž může uživatel nastavovat parametry dalším třídám, které s nimi dále pracují.

Ve Formu jsou implementovány tři panely nazvané *Měřicí deska*, *Čerpadlo* a *Průtokoměry*. Na každém panelu je implementován textchangerbox. Pomocí něj si uživatel vybere číslo sériového portu COM, na kterém bude chtít provádět sériovou komunikaci. Stisknutím tlačítka se nastaví příslušné třídě sériový port. V případě, že si uživatel vybere již používaný port, nebo port na který není nic připojeno, program napíše chybovou hlášku a nenastaví žádaný port a musí si vybrat jiný.

Dále je implementován překlikávací panel. Je rozdělen na několik záložek. V první záložce vyplní uživatel programu několik informací o průtokoměrech. Jsou jimi výrobce, typ a výrobní číslo průtokoměru. Tyto informace se využívají při vytvoření protokolu o kalibraci přístroje.

Druhou záložkou je Měření. V této záložce si uživatel může nastavit průtok čerpadla ve zvolených jednotkách. Průtok se dá nastavit buď posuvníkem, nebo vložením žádané hodnoty do políčka s hodnotou průtoku a stisknutím tlačítka nastavit. Hodnotu průtoku stanovuje čerpadlo. Čerpadlo je připojeno k měřicí ústředně, z které se pomocí D/A převodníku a zesilovače řídí pomocí změny napěťové úrovně. Pro převod je použit 12ti bitový převodník, který je integrován v mikroprocesoru měřicí ústředny.

Nastavení časového intervalu slouží k nastavení času, po který bude probíhat měření. To slouží k určení rozdílů mezi signály z průtokoměrů. Nastavení se provádí v tabulce *Čas měření*, přičemž první číslo udává počet minut a druhé počet vteřin. Samotné měření se spustí stisknutím tlačítka *Start*. Současně se spuštěním se začne odpočítávat čas zbývajících do konce, který se zobrazuje v políčku *Zbývajících čas*. Po zahájení se zakáže stisknutí tlačítka *Start* a tlačítka pro nastavení sériové komunikace (Odpojit). Je tomu tak z důvodu, aby nemohlo dojít ke znovuspuštění

měření, nebo ztrátě dat při přenosu. Po vypršení času určeného pro měření se tlačítka opět povolí a je možné měření opakovat.

The screenshot shows a software interface for flow measurement. At the top left, under 'Měřicí deska', there is a green 'Připojit' button and a 'COM 1' dropdown menu. To the right, under 'Průtokoměry', there is a 'COM 3' dropdown menu and a 'Nastavit' button. Below these are four tabs: 'Průtokoměry', 'Průtoky', 'Ověřování', and 'Kalibrace'. The main area contains a 4x4 grid of 16 flowmeter display boxes, each labeled 'Průtokoměr 1' through 'Průtokoměr 16'. To the right of this grid is a 'Jednotka průtoku' section with four radio buttons: [m3/hod] (selected), [l/hod], [l/min], and [l/s]. Below this is a 'Měřicí proces' section with a large green 'Start' button. Underneath the 'Start' button is a 'Nastavení času měření' section with two input fields: '0' for minutes and '5' for seconds. Below these is a 'Zbývajcí čas' display showing '0 : 00'. At the bottom is a 'Nastavení průtoku' section with a slider between 'Minimum' and 'Maximum' and a 'Nastavit čerpadlo' button. To the right of the slider is a '0' input field and a 'Nastavit' button.

Obr. 11 – Form, grafické rozhraní

Během měřicího procesu se v další záložce s názvem *Průtoky* zobrazují průběžné hodnoty průtoků. *Průtoky* jsou aktualizovány s určitou periodou přibližně 5s. Aktuální hodnota průtoku je vždy zobrazena v kolonce okamžitý průtok. Nad touto

kolonkou je umístěn sloupec, ve kterém je zobrazena historie přijatých pulsů. Po zahájení nového měření jsou hodnoty smazány.

Následující záložka slouží k samotné kalibraci přístrojů. Na levé straně záložky je umístěn panel, na kterém je implementováno dvakrát šestnáct textboxů. V prvním sloupci je zobrazen celkový počet přijatých pulsů od průtokoměrů za celý měřicí proces. V druhém sloupci je zobrazena difference, což je rozdíl, o kolik se liší hodnota nezkalibrovaných měřidel vůči etalonu. Etalon musí mít samozřejmě diferenci nulovou. Pro dobrou vizualizaci jsou jeho hodnoty označeny žlutou barvou. Pozice průtokoměru se nastavuje v sekci etalon nastavením čísla označující pozici příslušného měřidla. V sekci *kalibrace* se nastaví číslo průtokoměru, který chce uživatel kalibrovat. Stisknutím tlačítka *Získat* si program vyžádá aktuální konstanty průtokoměru, z kterých spočítá nové konstanty a nabídne je ke kalibraci. Stisknutím tlačítka *Zkalibrovat* se danému průtokoměru nastaví nové konstanty a kalibrace je dokončena.

Po zkalibrování přístrojů je nutné provést ověření kalibrace. K tomu slouží poslední záložka. Zde je za potřeby vybrat etalonový průtokoměr, pokud již nebyl vybrán. A spustit *ověřování*. *Ověřování* probíhá na třech průtocích, které si uživatel může zvolit. Defaultně je nastaven průtok 30, 60 a 90% výkonu čerpadla. Při každém z nastavených průtoků naměří ústředna dva tisíce pulsů na etalonovém průtokoměru. V tabulce *rozdíly pulsů* se poté zobrazí difference, která vyjadřuje rozdíly naměřených hodnot. Rozdíly by měly být v ideálním případě nulové. Po provedení ověřování si uživatel může vygenerovat protokol o kalibraci, ve kterém jsou uvedeny informace o kalibrovaném měřidle, o měřidle, kterým byla kalibrace provedena a o kvalitě kalibrace.

3.3.2 Informace o měřidlech

Průtokoměr	Výrobce	Typ	Výrobní číslo	Impulsní číslo
1	SENZORY CZ	SenzoFlow 40P	2014-01.001013	1000
2	SENZORY CZ	SenzoFlow M5000	2013-07.090812	1000
3				
4				
5				

Obr. 12 – Informace o měřidlech

Tato třída schraňuje informace o kalibrovaných a kalibračních průtokoměrech, které jsou ke kalibrační lince připojeny. Tyto informace jsou důležité pro vytvoření kalibračního protokolu, kterým měřidlo udává, s jakou přesností dokáže měřit. Informace o měřidle vyplní uživatel v panelu *Průtokoměry* podle toho, v jakém pořadí jsou měřidla k lince připojena. Těmito údaji je výrobce, typ a výrobní číslo měřidla.

Obsah položky výrobce označuje instituci, distributora nebo firmu, která daný průtokoměr zkonstruovala nebo ho vydává za vlastní výrobek. Typ průtokoměru označuje o jaký typ průtokoměru se jedná, například jde-li o indukční, nebo lopatkový průtokoměr. Každý výrobce má však vlastní značení. Výrobní číslo je nezaměnitelným identifikátorem výrobku. Každý výrobce by měl výrobní číslo použít pouze jednou. Výrobní číslo se používá při evidenci a revizích.

3.3.3 Měření

Třída s názvem *Měření* má za úkol řídit měřicí proces, tedy řídit zpracovávání signálů v měřicí ústředně. V uživatelském rozhraní Form je umístěno tlačítko *Start* a terminál pro nastavení délky měření. Uživatel dá stisknutím tlačítka *Start* impuls třídy, která vykoná následující operace. Jsou zde implementovány dva časovače. První časovač má periodu 1 000 ms a slouží jako hodiny. Druhý má periodu 5 000 ms a slouží jako časovač pro aktualizace.

Po stisknutí tlačítka *Start* dojde k předání dvou parametrů, kterými jsou dvě proměnné, které udávají čas měření, jsou to *minuty* a *sekundy*. Hodnota s názvem *minuty* se vynásobí šedesáti a přičte se k *sekundám*, čímž je nastaven odpočet hodin. V dalším kroku dojde k zakázání stisknutí tlačítek *Start* a *Připojení/Odpojení* sérového kanálu, aby nedošlo ke ztrátě dat. Současně se spustí odpočet hodin a podá se požadavek třídě *SP deska*, ke spuštění měřicího procesu. Pokaždé, když v časovači označeném jako hodiny dojde k přetečení, proměnná *sekundy* se dekrementuje. Je-li větší než nula, měření pokračuje dál a dojde pouze k updateu zbývajících času. Proměnná *sekundy* se vydělí 60, tak se získají zbývajících minuty a zbytek po dělení jsou zbývajících sekundy. Pokud ne, třídě *SP deska* dostane žádost o ukončení měřicího procesu. S časovačem hodiny je spuštěn i časovač pro aktualizování měřených hodnot, který ve svém přetečení požádá třídu *SP deska* o získání naměřených pulsů.

Po vypršení limitu dojde k zastavení obou časovačů a opětovné povolení tlačítek *Start* a *Připojení*.

3.3.4 Komunikace s měřicí deskou

Třída, která se zabývá komunikací mezi obslužným programem a měřicí deskou se nazývá *SPdeska*. Je v ní implementován objekt určený pro práci se vstupně výstupními porty. Nazývá se *Serial Port*. Pro správnou funkčnost je nutné nastavit mu několik parametrů. Tyto parametry musí být stejné jak na straně přijímače, tak na straně vysílače. Modulační rychlost sériového kanálu je nastavena na 115 200 Baud, počet přenášených bitů je 8, jméno portu se nastavuje ve třídě *Form*.

Je zde implementováno několik veřejných a privátních metod. První privátní metodou je *odesliZnak*. Tato metoda přímo spolupracuje se sériovým kanálem. Je v ní použita jeho funkce *Write*. Tato funkce odešle po sériovém kanálu jeden byte. Metodu *odesliZnak* využívá několik dalších veřejných metod pro řízení měřicí desky. Takovými metodami jsou *spustMereni*, *ukonciMereni*, *ziskejCounterArray*, *ziskejPulsArray*, *ziskejPeriodArray*.

Nejdůležitější věcí, kterou tato třída obsahuje, je přerušení od sériového kanálu. Toto přerušení nastane vždy, když program zaznamená příznak přenosu. Přenášená data jsou ukládána do *ReadBufferu*. Po přijetí se data načtou do lokálního jednorozměrného pole typu *byte* o velikosti *ReadBufferSize*. Tato proměnná udává kolik bytů bylo přijato a přesně na tuto hodnotu se inicializuje pole, do kterého se data uloží. Po naplnění pole je nezbytné rozhodnout o tom, zda jsou přijatá data věrohodná nebo při přenosu došlo k chybě. Při rozhodnutí o věrohodnosti hraje hlavní roli hned několik aspektů. Díky použití přenosového protokolu musí přijít alespoň tři informační byty. Z toho vyplývá, že když je *ReadBufferSize* menší než tři, nemá cenu provádět další testy, protože přijatá data jsou stoprocentně nevěrohodná. Dalším takovým aspektem je první byte z přijatých dat. Ten vždy udává počet vyslaných bytů. Je-li počet přijatých bytů stejný jako počet odeslaných, data by mohla být věrohodná. Jako poslední určení věrohodnosti se provede kontrolní součet se stejným algoritmem jako na straně odesílatele. Kontrolní součet je exkluzivní logický součet XOR všech odeslaných bytů, kromě posledního, kterým je právě *CRC*. Je-li hodnota vypočítaného *CRC* rovna přijatému *CRC*, jsou přijatá data s největší pravděpodobností věrohodná. V opačném případě je nezbytné opakované vyžádání dat.

V případě, že jsou přijatá data pravdivá, je třeba z nich vytvořit šestnáctibitové

proměnné, dva byty. Proměnná se složí ze dvou bytů. Spodní byte je každý sudý z pole hodnot. Vrchní byte je každý lichý předchozí. Tyto proměnné jsou vkládány do jednorozměrného pole integerů. Po zpracování všech dat se pole předá třídě zpracování dat, která s daty dále pracuje.

3.3.5 Zpracování dat

Tato třída se zabývá zpracováním a zveřejňováním obdržených dat od ostatních tříd. Ústředna během měřicího procesu vygeneruje několik typů dat, které se zde zpracovávají. Prvním typem zpracovávaných dat je pole celých pulsů. Toto pole informuje, kolik vygeneroval konkrétní průtokoměr period signálu za určitý čas. Tento typ se využívá pro získání kalibračních konstant a pro ověřování zkalibrovaných měřidel. Další typ dat byl nazván zpoždění signálu. Jedná se o jednorozměrné pole proměnných typu bezznaménkových integerů. Hodnota vyjadřuje počet vzorkovacích period od spuštění měřicího procesu po první náběžnou hranu vzorkovaného signálu, nebo počet period vzorkovacího signálu od ukončení měřicího procesu po první náběžnou hranu vzorkovaného signálu. Posledním typem dat je délka periody vzorkovaného signálu příslušející danému průtokoměru. Délka periody je vyjádřena počtem period vzorkovaného signálu za dobu mezi dvěmi náběžnými hranami vzorkovaného signálu.

V této třídě jsou inicializovány tři proměnné typu pole integerů pro zaznamenávání celých pulsů, zpoždění a perioda. Příchozí data typu celé pulsy jsou rovnou uloženy do proměnné vyjadřující celé pulsy. Toto pole slouží jako buffer pro aktualizaci okamžitých průtoků. Po přijetí jsou vynásobeny impulsní konstantou a zobrazena v záložce průtoky jako aktuální průtoky. Pokud přijdou data typu délka periody, uloží se rovnou do příslušného pole. Data typu zpoždění signálu mohou být dvou typů. Jedná-li se o počáteční zpoždění, jsou data rovnou uložena do příslušného pole. Jedná-li se o konečné zpoždění, jsou hodnoty dat konečného zpoždění odečteny od délky period vzorkovacího signálu. Touto matematickou operací se získá požadovaná hodnota, která je poté přičtena k dosavadní hodnotě zpoždění.

Pro zaznamenání celkového počtu načítaných pulsů je inicializováno pole typu double. Do tohoto pole jsou vždy přičteny hodnoty příchozích průběžných celých

pulsů, které přinášejí informaci o aktuálním průtoku. Po ukončení měřicího procesu se k těmto hodnotám přičte ještě hodnota zpoždění. Ta je vyjádřena počtem period vzorkovacího signálu, nikoli vzorkovaného signálu. Pro převedení na společnou jednotku, je třeba zpoždění vydělit proměnnou vyjadřující délkou periody, která určuje, kolik period vzorkovacího signálu připadá na periodu vzorkovaného. Zpoždění by potom mělo mít hodnotu od 0 do 2 pulsů, neboli period vzorkovaného signálu. Díky tomuto způsobu lze určit počet period měřeného signálu na desetiny periody.

3.3.6 Ověřování

Ověřování je proces, kterým se testují průtokoměry na různých průtocích. Uživatel si zvolí různé průtoky, na kterých chce měřidla testovat a požadovaný objem, který má během testování průtokoměry protéci. Alespoň jeden z testovaných průtokoměrů je etalon, vůči kterému se porovnávají průtoky ostatních měřidel. Pro testování byla v obslužném programu vytvořena záložka, která je určena pro ověřovací proces.

The screenshot shows the 'Ověřování' (Verification) tab of a software interface. It contains several input fields and three data tables.

Input Fields:

- Ověření:**
 - Etalonový průtokoměr: 1 (dropdown)
 - DN: (dropdown)
 - Kalibrační objem 1: 0,5 [m3]
 - Kalibrační objem 2: 0,5 [m3]
 - Kalibrační objem 3: 0,5 [m3]
 - Import ze souboru: (button)
 - Průtok 1: 0 (input), 0 (input) [pulsy]
 - Průtok 2: 0 (input), 0 (input) [pulsy]
 - Průtok 3: 0 (input), 0 (input) [pulsy]
 - ☒ l/min ☐ m3/min
 - Export do souboru: (button)
 - Spustit ověření: (button)
- Protokol:**
 - C:\Users\admin\Documents (text field)
 - Vytvořit protokol o měření: (button)

Data Tables:

Průtok 1		
č.	V	δ [%]
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0

Průtok 2		
č.	V	δ [%]
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0

Průtok 3		
č.	V	δ [%]
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0

Obr. 13 – Ověřování průtokoměrů

Ověřování je postaveno na principu čítání pulsů generovaných průtokoměry. Jeden z průtokoměrů je určen jako etalonové měřidlo. Vůči tomuto měřidlu se porovnávají počty pulsů ostatních měřidel. Výsledkem ověřování je difference pulsů, která by měla být v ideálním případě nulová a chyba oproti etalonu stanovená v procentech. Ověřování provádí měřicí ústředna. V obslužném programu se pouze určí parametry, jakými jsou průtoky, na kterých se budou průtokoměry testovat, množství

kapaliny protečené linkou během testu, a který z průtokoměrů je etalon.

Uživatel zadá žádaný objem nebo počet pulsů, který musí linkou protéct během testu. Při měření se sleduje počet načítaných pulsů etalonového průtokoměru. Počet pulsů je úměrný protečenému objemu média. Když je vyšší než žádaná hodnota, může být měření ukončeno a výsledky testu mohou být zpracovány. Stejně měření se opakuje na každém zvoleném průtoku.

Výsledkem testování je tabulka, ve které je uvedeno, jaké množství kapaliny proteklo měřidlem a rozdíl o kolik se tato hodnota liší oproti etalonu. Tyto hodnoty jsou použity pro přepočet kalibračních konstant, díky kterým se dají průtokoměry zkalibrovat.

3.3.7 Kalibrace

Kalibrace je proces, který se snaží nastavit měřidlo tak, aby ukazovalo co nejpřesnější hodnotu. Měřidlo musí ukazovat naměřenou hodnotu s určitou přesností. Tato přesnost je udána tolerancí, kterou udává výrobce.

Pro kalibraci průtokoměrů se využívá hodnot získaných *ověřením*. Pomocí tohoto programu lze kalibrovat průtokoměry, které využívají komunikační protokol SIMPLE. Pomocí tohoto protokolu komunikují například průtokoměry firmy Limesa. Tento protokol je určen pro komunikaci přes rozhraní RS232.

Obr. 14 – Kalibrace průtokoměrů

Obsluha linky připojí průtokoměr, který se bude kalibrovat k počítači přes převodník USB/RS232. Poté si v panelu *Form* vybere port, na kterém bude komunikace probíhat. V záložce *Kalibrace* se nastaví průtokoměr, který se bude kalibrovat.

Pro výpočet nových kalibračních konstant je potřeba získat současné konstanty. Ty si lze z průtokoměru vyžádat. Po zaslání požadavku průtokoměru je průtokoměr navrátí zpět. Konstanty se zobrazí v políčkách *old K* a *old P*. Tyto konstanty ovlivňují generování pulsů. Popisují rovnici přímky $y = K \cdot x + P$, kde K je směrnice přímky a P je *offset*, neboli posun po ose y .

$$new K = \frac{old K \left(FL_2 \frac{100 - \delta_2}{100} - FL_1 \frac{100 - \delta_1}{100} \right)}{FL_2 - FL_1}$$

Vzorec 4 – Výpočet konstanty *new K*

Pro výpočet nové konstanty *new K*, nové směrnice, je potřeba znát současnou směrnici přímky *old K*, dva různé průtoky linkou, při který bylo prováděno ověřování FL_1 a FL_2 . δ_1 a δ_2 , udávají rozdíl hodnoty naměřené daným průtokoměrem oproti hodnotě naměřené etalonem při daném průtoku. Rozdíl průtoků FL_1 a FL_2 by měl být řádový. Tuto novou kalibrační konstantu lze vypočítat podle vzorce 7.

$$new P = FL_2 \frac{100 - \delta_2}{100} - new K \frac{FL_2 - old P}{old K}$$

Vzorec 5 – Výpočet konstanty *new P*

Konstanta *new P*, *offset* přímky, se určí pomocí vzorce 8, kde FL_2 vyjadřuje průtok, δ_2 rozdíl naměřené hodnoty průtokoměru vůči etalonu, novou směrnici *new K*, starou směrnici *old K* a starou konstantu *old P*.

Nové hodnoty se zobrazí v políčku *new K* a *new P*. Kalibrace proběhne tak, že se konstanty *old K* a *old P*, které jsou nastaveny v průtokoměru přenastaví nově určenými konstantami *new K* a *new P*. K přenastavení konstant se využije protokolu SIMPLE.

3.4 Komunikace s průtokoměrem

Pro komunikaci s průtokoměry se z pravidla využívá sériový kanál. Samotný sériový kanál slouží pouze k posílání znaků. Pro komunikaci se zařízeními je výhodné používat přenosové protokoly. Pro ukázkou je zde rozebrán protokol s názvem SILMPLE, který pro komunikaci s průtokoměry používá firma Limesa.

Data jsou posílána v paketech. Příchozí a odchozí pakety, z hlediska průtokoměru, se liší svou strukturou a délkou. Příchozí paket bývá nejčastěji vyslán z počítače, nebo z nějakého zařízení, které potřebuje z průtokoměru získat konkrétní data, aby mohla být dále zpracována. První byte paketu adresovaného pro průtokoměr oznamuje délku zprávy. Délka zprávy informuje o počtu bytů, které může průtokoměr očekávat.

Následujících n bytů je požadavek pro průtokoměr. Tento požadavek oznámí průtokoměru, jaká data chce odesílatel poslat. Pro každý průtokoměr je vytvořena tabulka požadavků. Požadavkem může být například protečený objem média, okamžitý průtok, identifikační číslo průtokoměru a podobně.

Předposlední byte je standardně konstantní. Má hodnotu 0 hexa. Poslední byte slouží jako kontrolní součet, který je nazván jako CHSUM. Do kontrolního součtu jsou zakomponovány všechny byty paketu, což zvyšuje věrohodnost přenášení informace.

$$CHSUM = NOT(byte[0] XOR byte[1] XOR ... XOR byte[n]) + 0x01$$

Vzorec. 6 – Výpočet kontrolního součtu pro komunikaci s průtokoměrem



Obr. 15 – Protokol komunikace s průtokoměrem

Paket odesílaný z průtokoměru má stejnou strukturu jako paket adresovaný pro průtokoměr. Jediné v čem se pakety liší je jejich délka.

3.5 Řízení oběhového čerpadla

Pro vytvoření průtoku média kalibrační linkou je použito oběhové čerpadlo firmy Grundfos. Maximální průtok tohoto čerpadla je 23 m³/hod. Čerpadlo je řízeno frekvenčním měničem AQUA Drive od firmy Danfoss. Měnič byl řízen napětím na potenciometru, takže ovládání průtoku bylo pouze manuální.

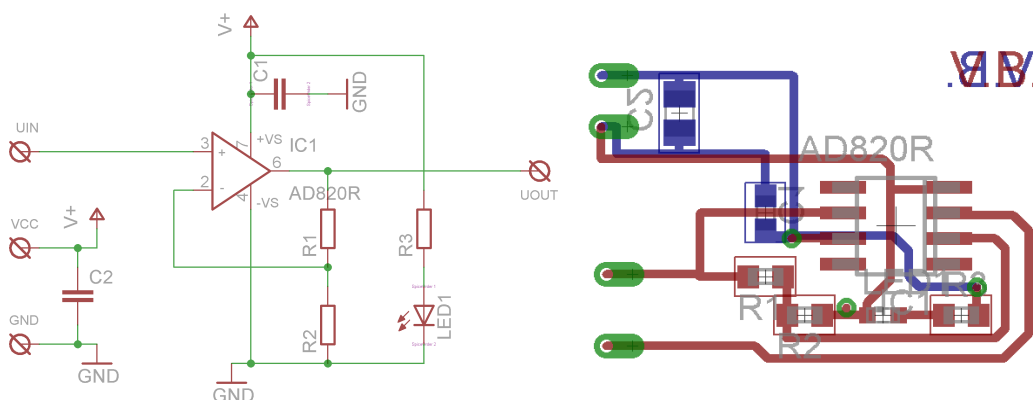
Úkolem je řídit čerpadlo automaticky, respektive pomocí obslužného programu. K tomu je využit digitálně analogový převodník DAC0, který je integrován v mikroprocesoru měřicí ústředny. Vstupní napětí frekvenčního měniče je 0 až 10 V. Výstupní napětí DAC0 je 0 až 2,4V. Maximální hodnota výstupního napětí převodníku je 2,4 V, což je přibližně čtyřikrát méně, než je řídicí napětí měniče, proto musí být napětí zesíleno. Pro zesílení napětí byl použit operační zesilovač AD 820. Jeho výhodou je, že má funkci rail-to-rail, která umožňuje, že se saturační napětí OZ U_{SAT} může blížit napájecímu napětí U_{CC} . OZ je zapojen jako neinvertující zesilovač. Výhodou neinvertujícího zesilovače je vysoký vstupní a nízký výstupní odpor a neotáčení fáze signálu. Zesílením operačního zesilovače se spočítá jako poměr výstupního napětí ku vstupnímu podle *vzorce 7*. A resistory se poté volí podle *vzorce 8*.

$$A_u = \frac{U_{out}}{U_{in}}$$

Vzorec 7 - Definice zesílení

$$\frac{R_2}{R_1} = A_u - 1$$

Vzorec 8 - Návrh rezistorů pro neinvertující zesilovač s OZ



Obr. 14 – Schéma zapojení a návrh DPS zesilovače pro řízení čerpadla

Výkon čerpadla, respektive hodnota D/A převodníku se nastavuje pomocí obslužného programu. V panelu *Měření* je několik radiobuttonů, díky kterým si lze zvolit jednotku žádaného průtoku. Pro samotné nastavení průtoku lze použít posuvník, který má 2^{12} hodnot, nebo hodnotu lze nastavit přímo pomocí políčka ve kterém je zobrazena žádaná hodnota průtoku. Po stisknutí tlačítka *Nastavit* se hodnota žádaného průtoku odešle do ústředny. Tato hodnota se odesílá ve 3 bytech. První byte upozorní ústřednu, že se bude nastavovat hodnota DAC0. Další dva byty určují výkon čerpadla (vyšší a nižší byte). Po přijetí žádané hodnoty se provede nastavení D/A převodníku, kde se nižší byte do registru DAC0L a vyšší byte zapíše do registru DAC0H. Převodník je nastaven tak, že se jeho hodnota mění společně se změnou hodnoty registru DAC0H.

Po nastavení registrů se na výstupu D/A objeví napětí, které je zesíleno a přivedeno na vstup frekvenčního měniče, který na základě napětí řídí výkon oběhového čerpadla.

Závěr

Cílem této bakalářské práce bylo vytvořit program, který bude obsluhovat měřicí ústřednu umístěnou v měřicí soustavě pro kalibraci až šestnácti průtokoměrných měřidel. Současně s vytvořením programu pro měřicí ústřednu bylo vytvoření obslužného programu do počítače, který bude měřicí ústřednu obsluhovat a zpracovávat data z ní obdržená.

Ke splnění těchto bodů bylo třeba nastudovat problematiku mikroprocesorů řady 8051.

Pro měřicí ústřednu byl vytvořen software, který sbírá data na základě signálů generovaných průtokoměry, z nichž je možné určit okamžitý průtok. Software umožňuje sbírat data pro výpočet a korekci kalibračních konstant a sběr dat pro ověřování přesnosti měřidel a pomocí sériového kanálu komunikuje s počítačem.

Pro jednoduchou obsluhu měřicího systému byl sestaven počítačový program ve vývojovém prostředí Visual studio 2010 Express v jazyce C#. Program má Grafické uživatelské rozhraní GUI (Graphic User Interface), díky kterému je ovládání daleko snazší, než by bylo ovládání z konzole. Obslužný program posílá přes RS232 požadavky ústředně, jako například *spust' měření*, *spust' ověřování*, *ukonči měření*, *vrat' data*, atd. Dále má za úkol vyžádaná data zpracovávat, zobrazit na monitoru, uložit kalibrační protokol a navrhnout nové kalibrační konstanty, je-li to nutné.

Měřicí ústředna je řízena mikroprocesorem, to znamená, že data jsou zpracovávána sekvenčně. Zdá se, že daný procesor při daném pracovním kmitočtu pracuje na hranici svých možností. Pro podobné zpracování více signálů by bylo možná výhodnější zpracovávat signály pomocí hradlového pole, jelikož hradlová pole umožňují zpracovávat signály paralelně. Mikroprocesor by se poté dal využít na složitější operace. Dalším řešením by bylo použití vícebitového procesoru.

Seznam použité literatury

- [1] PLÍVA Z., DRÁBKOVÁ J. *Metodika zpracování diplomových, bakalářských a vědeckých prací na FM TUL*, Liberec 2007, ISBN 978-80-7372-189-3
- [2] VINŠ J. *Návrh průtokoměrné linky pro testování bytových a průmyslových průtokoměrů*. Liberec, 2007. 58 s. Bakalářská práce. Fakulta mechatroniky, informatiky a mezioborových studií. Technická univerzita v Liberci. Vedoucí diplomové práce Ing. Lubomír Slavík , Ph.D.
- [3] Jelínek J. *Měření průtoku* [online], přednášky předmětu MT2 Fakulty mechatroniky, informatiky a mezioborových studií TU v Liberci. Liberec, 2014. 446 – 565s. URL: <https://elearning.fm.tul.cz/course/view.php?id=37> .
- [4] HEROUT P. *Učebnice jazyka C 1. díl*, České Budějovice, 2010, ISBN 978-80-7232-383-8
- [5] HEROUT P. *Učebnice jazyka C 2. díl*, České Budějovice, 2010, ISBN 978-80-7232-367-8
- [6] SKALICKÝ P. *Mikroprocesory řady 8051* 2. vydání, Praha, 1998, ISBN 80-86056-39-2

Seznam příloh

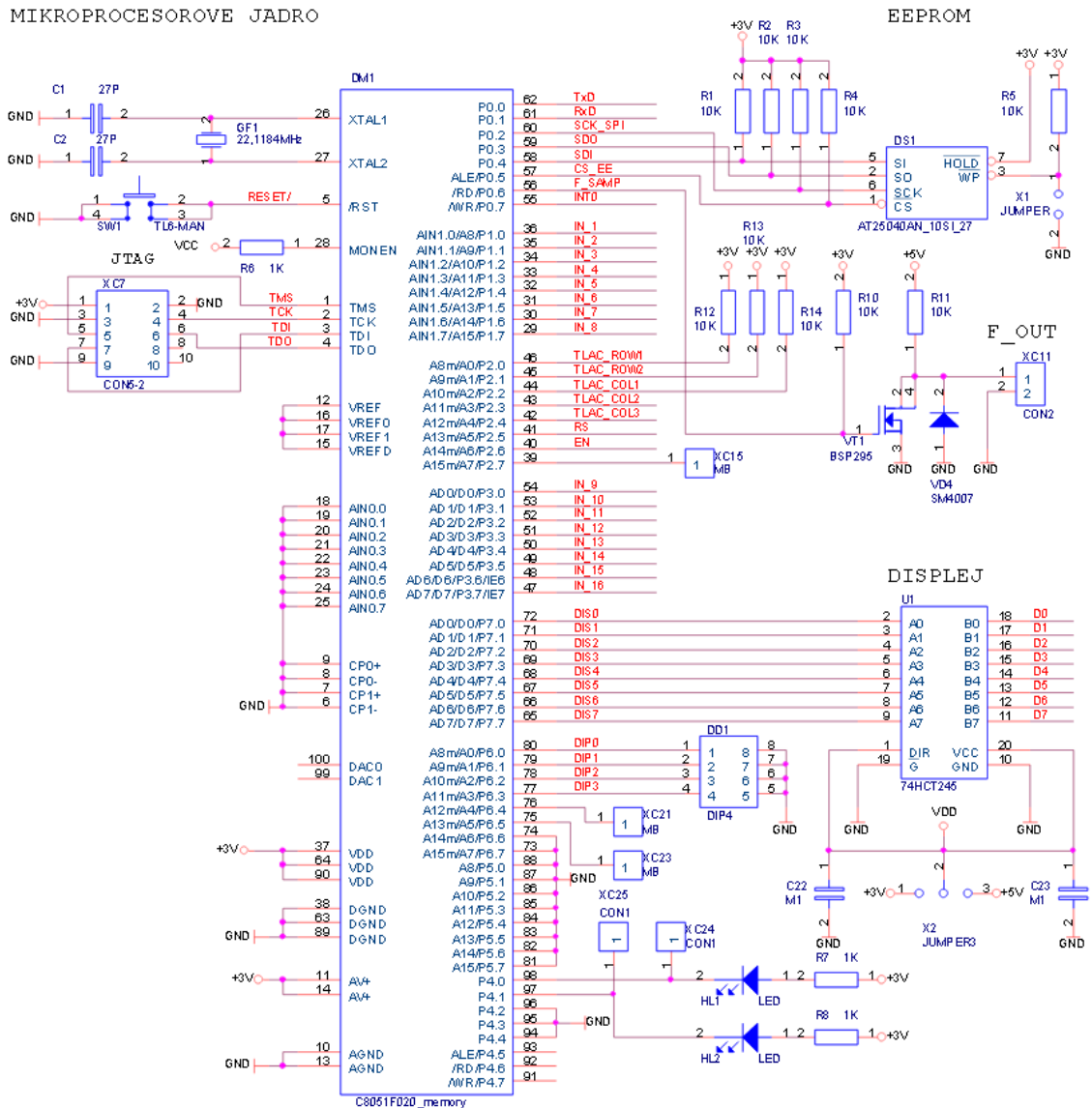
Příloha A - Kalibrační linka.....	50
Příloha B - Ukázka návrhu měřicí desky.....	51
Příloha C - Měřicí deska.....	52
Příloha D - Ukázka kódu v jazyce C.....	53
Příloha E - Ukázka kódu v jazyce C#.....	54

Příloha A Kalibrační linka



Příloha B Ukázka návrhu měřicí desky

MIKROPROCESOROVE JADRO



Příloha C Měřicí deska



Příloha D Ukázka kódu v jazyce C

```
void Compare(U16 pulsVector)
{
    char i;
    for(i = 0; i<Q; i++){
        if(((pulsVector>>i)&0x1)>0){
            counterArray[i]++;

            if(counterArray[i]<2)
            {
                periodArray[i] = delayTime;
            }
        }
    }
}

void Finish(U16 pulsVector)
{
    char i;
    for(i = 0; i<Q; i++){
        if(((pulsVector>>i)&0x1)>0){
            if(periodArray[i] > 1)
            {
                periodArray[i] = delayTime;
            }
        }
    }

    if(delayTime >= 4500){
        startMode = 3;
    }
}
```

Příloha E Ukázka kódu v jazyce C#

```
public void start() // provede se se stisknutím tlačítka start
{
    cKalibrace.start();
    clearPulsy();
}
private void clearPulsy()
{
    for (int i = 0; i < namerenePulsy.Length; i++)
    {
        namerenePulsy[i] = 0;
    }
}
public void nastavJednotku(int jednotka)
{
    this.jednotka = jednotka;
}
private void vycistit(object sender, EventArgs e)
{
    for (int i = 0; i < 16; i++)
    {
        richTextBoxy[i].Clear();
    }
}
public ZpracovaniDat(RichTextBox[] richTextBoxy, Kalibrace cKalibrace)
{
    this.richTextBoxy = richTextBoxy;
    this.cKalibrace = cKalibrace;
}
```